ARL-TN-25

AR-006-666

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Technical Note 25

SIX DEGREE OF FREEDOM FLIGHT DYNAMIC MODEL OF A MK-82 STORE

by

M.A. SHILO

Approved for public release

JANUARY 1994

AR-006-666

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Technical Note 25

SIX DEGREE OF FREEDOM FLIGHT DYNAMIC MODEL OF A MK-82 STORE

by

M.A. SHILO

SUMMARY

A six degree of freedom flight dynamic model incorporating an aerodynamic database derived from DSTO wind tunnel data was developed for the Mk-82 store. The model can be linked with aircraft flight dynamic models, and forms part of a general programme to model aircraft carriage and delivery, weapon release and store ballistics. A rudimentary ejection model was also incorporated to represent the forces and moments experienced by the store at the point of release. The model is valid for free stream flight only, and ignores the effects of aircraft-store interaction.

DSTO
AUSTRALIA

POSTAL ADDRESS: Director, Aeronautical Research Laboratory
506 Lorimer Street, Fishermens Bend
Victoria 3207, Australia.

TABLE OF CONTENTS

Page Nos.

List of Figures

NOTATION

a	Variable of arbitrary value
α	Angle of attack or incidence in degrees
b	Store reference diameter
β	Angle of sideslip
C_X	Non dimensional force coefficient, x direction
C_Y	Non dimensional force coefficient, y direction
C_Z	Non dimensional force coefficient, z direction
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient
C_n	Yawing moment coefficient
C_{l_p}	Rolling moment derivative due to rate of roll
C_{m_q}	Pitching moment derivative due to rate of pitch
C_{n_r}	Yawing moment derivative due to rate of yaw
C_{n_p}	Yawing moment derivative due to rate of roll
C_{y_p}	Side force derivative due to rate of roll
CG	Centre of gravity
I_{xx}	Moment of inertia about the x axis
I_{yy}	Moment of inertia about the y axis
I_{zz}	Moment of inertia about the z axis
j	Imaginary number
ω	Frequency
M	Mach number
p	Rate of roll
ϕ	Roll angle
q	Rate of pitch
$QBAR$	Dynamic pressure
r	Rate of yaw
S	Reference area
θ	Pitch attitude
u	Body-axes X velocity
v	Body-axes Y velocity
V	Airpath velocity vector, true air speed
w	Body-axes Z velocity
x	Longitudinal axis (+) forward
y	Lateral axis (+) starboard
z	Vertical axis (+) down
ζ	Damping ratio

Subscripts

s	Sine basis for calculation of α and β
t	Tangential basis for calculation of α and β

1 INTRODUCTION

Under Air Force Research Requirement AFRR 7/90 'F-111C/RF111-C Modelling', DSTO is providing support to the RAAF in the verification of store release and delivery accuracies of the weapons systems in the F-111C Avionics Update Programme.

The accuracy of store delivery depends on a number of factors, including aircraft release conditions, store ejector characteristics, aerodynamic release disturbances, store aerodynamic characteristics and atmospheric disturbances. The calculation of release point should ideally account for these errors.

As part of the process of store delivery prediction and verification DSTO is developing dynamic models which can simulate the store carriage and release, and store ballistic behavior of unguided weapons. Information for use in these models is being derived from available data reports, and from wind tunnel and flight test measurements.

In this report the development of a flight dynamic model for the Mk-82 store which can be linked with models of aircraft flight behavior is described. The model is valid for free-stream flight only, but is structured to enable the effects of aerodynamic release disturbances, detailed ejector characteristics, and atmospheric disturbances to be incorporated.

2 FLIGHT DYNAMIC MODEL

The flight dynamic model of the Mk-82 store was developed using a standard Six Degrees of Freedom model of rigid body motion developed in Air Path Axes (Reference 1). The model was written using the Advanced Continuous Simulation Language (ACSL, Reference 2) at ARL Melbourne. The program uses a fourth order Runge-Kutta integration method, with quaternian parameters for the calculation of aircraft attitude, and direction cosines for the gravity vectors. The use of ACSL enables a range of dynamic system analysis tools to be used during model development and application. The information used in the simulation of the Mk-82 store is listed below.

1. The physical dimensions, mass and inertia properties of the Mk-82 store were necessary for body axes force and moment calculations (see Appendix A).

2. Body axes forces and moments were calculated in subroutine AERO2, which retrieves the appropriate aerodynamic coefficients from the wind tunnel database using database access routines which convert the aerodynamic forces from missile to airpath axes (see Appendix B).

3. A database of aerodynamic coefficients was incorporated in the form of data tables with independent variables Mach No, angle of attack, and roll angle. The original database, obtained from wind tunnel tests, covered only positive angles of attack and contained non-zero aerodynamic values at zero angle of attack and roll angle (see Appendix C).

4. Unlike an aircraft, the Mk-82 store cannot be trimmed using movable control surface deflections. The equilibrium condition of the modelled store is achieved by the addition of incremental body axes forces and moments, and the use of a non linear equation minimisation technique. This procedure is initiated by keying the command 'eigen'. With the store in equilibrium, standard linear dynamic analysis techniques available within

ACSL can be used to analyse the linear dynamic behavior, and provide information about the modes of oscillation and other motions. See Appendix D Example Simulation Run, step 12, for an explanation of the command 'eigen'.

5. Store ejection was simulated by the addition of the two subroutines, EJECT and EJECT2, which provide a rudimentary representation of store ejection forces from the outboard and inboard stations of a BRU-33/A Vertical Ejector Rack (see Reference 3, also see Appendix B). Ejection is achieved by applying fixed ejection forces based on measured data to the store model for 0.12 seconds.

6. The subroutine MAKEFILE was included to generate an ASCII file of the store trajectory position, velocity, and orientation for use in a graphic display of the behavior of the store flight behavior, using a program developed for a Silicon Graphics Workstation.

A more detailed discussion of the modifications and additions made to the aerodynamic databas? is presented in Appendix B.

3 THE MK-82 AERODYNAMIC DATABASE

3.1 Database Source

The Aerodynamic Database was obtained from two sources of experimental data, viz the DSTO wind tunnels at Salisbury, South Australia and Melbourne, Victoria. The Melbourne data were measured in the transonic wind tunnel, and covered the Mach number range of (approximately) 0.9 to 1.2 , with angle of attack (α) 0 to 30°, and roll angle (ϕ) $-45°$ to $+45°$. The tests were performed using a standard pitch/roll test rig with sting mounted strain gauges to measure the forces and moments. Note, with sting mounted models, combinations of angle of attack and roll angle are used to provide the required ranges of angle of attack and sideslip angle. Because the sting is attached to the movable test model, the resulting force and moment values are measured in body axes. However, the experimental results of both Melbourne and Salisbury tunnels were resolved into missile axes and incorporated into a new database with a different format.

Body axes and missile axes are defined as follows:

- Body axes are a set of orthogonal axes that are fixed to the body, with the x, y, z axes for axi-symmetric bodies aligned with the principal moments of inertia of the body.
- Missile axes are similar to body axes, but with only the x-axis aligned with the corresponding body axes moment of inertia, I_{xx}. The y-axis remains in the inertial x-y plane (horizontal) regardless of the motion of the body, such that body axes and missile axes differ by a rotation of ϕ about their common x-axis only. Figure 1 shows the difference between body and missile axes.

3.2 Original Format of Database

The database file contains the following data in 17 ASCII columns.

1. Data point number

2. Incidence set number
3. Roll angle set number
4. Mach set number
5. Angle of incidence, 0° to +30° (2° increments)
6. Store roll angle, −45° to +45° (7.5° increments)
7. Mach number, 0.4 to 1.2
8. C_X - non dimensional force coefficient, x direction
9. C_Y - non dimensional force coefficient, y direction
10. C_Z - non dimensional force coefficient, z direction
11. C_l - rolling moment coefficient
12. C_m - pitching moment coefficient
13. C_n - yawing moment coefficient
14. C_{l_p} - rolling moment derivative
15. C_{m_q}, C_{n_r} - pitching and yawing moment derivatives (a single value is provided for both variables: see section 3.3)
16. C_{n_p} - yawing moment derivative due to rate of roll*
17. C_{Y_p} - force derivative due to rate of roll*

* Since the contribution to resultant force and moment provided by the terms C_{n_p} and C_{Y_p} is not significant and the quality of the data for these terms is not known, these variables are not used in this simulation.

3.3 Assumption

Close examination of the database shows that values of the variable C_{m_q} for $\alpha = a, \beta = b$ are identical to those of C_{n_r} for $\alpha = b, \beta = a$. This is not strictly correct, but if the approximation that $\alpha_s = \alpha_t$ is used, the values in the case above are valid. The following simple comparison test demonstrates the insignificance of the error in the assumption over the range covered by the database:

α_t	β_s	α_t/α_s	β_t/β_s
2	1	0.99954	0.99999
5	1	0.99633	0.99999
10	1	0.98479	0.99999
20	1	0.93715	0.99998
30	1	0.85082	0.99999
2	2	0.99999	0.99999
5	5	0.99997	0.99997
10	10	0.99952	0.99954
20	20	0.99201	0.99294
30	30	0.95662	0.96723

where

$$tan\alpha_t = w/u$$

$$tan\beta_t = v/u$$

$$sin\alpha_s = w/V$$

$$sin\beta_s = v/V \text{ and}$$

u, v, w are the components of the air path velocity vector V, along the body axes x, y, z. Figure 2 shows the difference between α_t and α_s, and β_s and β_t.

Notice that the worst case error is for the case where $\alpha = 30°, \beta = 1°$, or $\beta = 30°, \alpha = 1°$, where the associated error is approximately 15% . For the remaining tabulated values, the error is less than 6.5% , indicating that the approximation $\alpha_t = \alpha_s$ and $\beta_s = \beta_t$ is reasonable over the range covered by the database.

3.4 Revised Database Format

A set of standard database handling routines developed by Air Operations Division at ARL for use in flight dynamic simulations is used for accessing the aerodynamic data.

The original database was changed into the format required by the standard database extraction routines. The new database has the following format:

```
title mk82cxdatabase
*
thruput cx  (machno,phi,theta)
thruput cy  (machno,phi,theta)
thruput cz  (machno,phi,theta)
thruput cl  (machno,phi,theta)
thruput cm  (machno,phi,theta)
thruput cn  (machno,phi,theta)
thruput clp (machno,phi,theta)
thruput cmq (machno,phi,theta)
*
machno varbpt 0.4,0.5,0.6,0.7,0.8,0.85,0.9,0.95,1.0,1.05,1.1,1.15,1.2/
phi    conbpt -45.0,45.0,7.5/
theta  conbpt 0.0,30.0,2.0/
*
CX     POINTS
-0.1556,    -0.1552,    -0.1524,    -0.1476,    -0.1472,
-0.1440,    -0.1394,    -0.1327,    -0.1264,    -0.1170,  etc.
```

The database subroutines will calculate the longitudinal and lateral force and moment coefficients corresponding to the flight conditions of Mach No, roll angle ϕ, and pitch attitude θ.

The **thruput** statements notify the subroutines that each force or moment coefficient is determined from the independent variables Mach No (abbreviated to **machno** in the program), **phi**, and **theta** information, where M, ϕ, θ are calculated in the main ACSL program or subroutines and are passed to the database access routines. The force and moment coefficients are called **functions**.

The **varbt** statement notifies the subroutine of all the values of **machno**, the values not being spaced equally apart. The **conbpt** statement allows the lower and upper bounds of the variables **phi** and **theta** to be stored with a constant interval, thus using memory economically.

The values of the database functions at the discrete values of the variables **machno**, **phi**, and **theta** are called **breakpoints**. The data are listed until all breakpoint combinations have been used. The ordering of the breakpoints is such that the function depending on the last variable varies first. For example, if a function C_c has two variables which have two breakpoints each, the correct order for the data to appear is :

$$(1,1),(1,2),(2,1),(2,2)$$

thus cycling through all of the possible combinations.

The original database was provided in an ASCII format file. The database modification was performed on an IBM PS/2 model 70 personal computer. A listing of the program used for the conversion is presented in Appendix C.

3.5 Database Bias

The aerodynamic data for the Mk-82 store as noted in Section 3.2 covers only positive angles of attack up to 30° and roll angles of $-45°$ to $+45°$. For the flight dynamic model, it is necessary to assume that the data are symmetrical about 0° angle of attack so that positive and negative changes in both angle of attack and sideslip can be simulated.

For a symmetrical aerodynamic shape at zero angle of incidence and zero angle of roll, it would be expected that pitching and yawing moments, together with vertical and side forces, should be zero.

Inspection of the database showed that non-zero values of pitching and yawing moments, as well as vertical and side forces, were measured at zero incidence and roll. The non-zero terms became evident when the simulation was run, resulting in non-linear behaviour at low angles of incidence. Figure 3 shows the limit cycle oscillations of α due to the non-zero bias in the database with the model performing pure pitching motion. Figure 4 shows the limit cycle oscillations of α when all the degrees of freedom are included in the model. To ensure continuity of the forces through zero angle of attack it is necessary to remove these biases.

The unmodified database caused incorrect eigenvalues to be calculated by the ACSL analysis capability and therefore caused unpredictable stability characteristics at low angles of incidence. Removal of the bias by linear subtraction caused the amplitude of the limit cycle to approach zero, and resulted in acceptable behavior, as illustrated in Figures 5 and 6, thus providing an acceptable solution. Figure 7 shows a plot of C_m versus θ, clearly showing the database bias.

It is recommended that the current database be replaced by a more accurate database when one becomes available. Further work is under way to improve the quality of the aerodynamic database.

4 PROGRAM STRUCTURE

The Mk-82 simulation program consists of a main ACSL program which contains the equations of motion and a set of FORTRAN subroutines which determine the aerodynamic forces and moments. Various additional subroutines provide necessary initial conditions and calculate information for graphical output. Figure 8 shows a flow chart of the program, while Figure 9 shows the block diagram of the air path axes system for the store motion.

5 VERIFICATION OF FLIGHT DYNAMIC MODEL

In order to determine the validity of the six degree of freedom model, a series of tests was performed on constrained versions of the model. The tests involved separating the lateral and longitudinal modes and analysing them individually. All tests utilised the revised database.

5.1 The Longitudinal Model

The first test involved constraining the six degree of freedom model to a three degree of freedom longitudinal model by removing the linear side force components, gravity, and the roll and yaw moment components of the full model. The reduced model could now move only in the x-z plane.

After trimming the store at $\alpha = 0.0^o$, an eigen analysis was performed to identify the dynamic modes of the reduced model. The eigen values were :

1. -0.00872699
2. $-0.2128 \pm 4.9489j$.

The first mode is a stable transient mode which corresponds to the drag force acting along the x-axis.

The second mode is a stable oscillatory mode which corresponds to an undamped natural frequency of $\omega = 4.95$ rad/s, and a damping ratio of $\zeta = 0.04$. This mode is characterised by a porpoise type of motion, with pitching and vertical translation oscillations occurring approximately 90^o out of phase.

To illustrate the oscillatory nature of the store motion, a simulation run was calculated with the initial condition of $\alpha = 5^o$. Figure 10 shows a plot of α, while Figure 11 shows the eigen vectors relating to the oscillatory eigen value.

> The transient mode is illustrated as a drag force because the eigen vectors indicate that velocity decreases as x increases.
>
> The oscillatory mode shows a simultaneous pitch oscillation and translational vertical oscillation, with θ, z, and α lagging pitch rate q by approximately 90^o.
>
> Note that β is always equal to zero in this case because the store may not move laterally.

The Mk-82 store is modelled assuming identical aerodynamic forces in pitch and yaw as noted in Sections 3.2 and 3.5 and so will have identical lateral and longitudinal dynamic properties. The results of the dynamic analysis are consistent with these assumptions. Further testing of the dynamic behaviour including the effect of rolling moments, inertia coupling, and non-linear aerodynamic effects should be undertaken to verify all aspects of the six degree of freedom flight dynamic model.

Figure 12 shows a plot of β for a simulation run with initial condition $\beta = 5^o$, and Figure 13 shows the eigen vectors corresponding to the oscillatory mode, along with a representation of the associated motion. This mode is characterised by a 'fish tailing' type of motion, with yawing and horizontal translations occurring approximately 90^o out of phase.

Figure 14 shows a plot of α versus time for a simulation run of the full six degree of freedom model released at an altitude of 1000 feet and initial angle of attack of $\alpha = 5.0^o$. The change in oscillatory behaviour at time t=3.5 seconds occurs when the roll rate coincides with the pitch and yaw frequencies and demonstrates the effects of roll coupling.

'Roll coupling' is the term used to describe instability caused by the dependence of pitching moment on both pitch and roll rate, and the dependence of yawing moment on both yaw and pitch rate.

These preliminary tests indicate that the six degree of freedom model behaves as expected. However, it is recommended that model verification and validation tests are performed as further data from wind tunnel, computational fluid dynamics and flight tests becomes available.

6 CONCLUSIONS

A six degree of freedom flight dynamic model has been developed for the Mk-82 store. The model includes a linear analysis capability which is useful in the understanding of the store motion. Preliminary tests indicate that the model is valid, but further testing is recommended.

The database used in the model was compiled from wind tunnel test data obtained from ARL Salisbury and ARL Melbourne wind tunnel tests. Bias terms have been removed from the data for use in the six degree of freedom simulation application. It is suggested that the current database be replaced with a more accurate database when one becomes available.

ACKNOWLEDGEMENTS

The advice and assistance of Mark Cooper and Colin Martin in the developement of the six degree of freedom flight dynamic model for the Mk-82 store and the compilation of this note is gratefully recognised.

REFERENCES

1. Gibbens, P. **A Flight Dynamic Simulation Program in Air-Path Axes using ACSL**, Aerodynamics Technical Memorandum 380, Aeronautical Research Laboratory, Melbourne, June 1986.

2. **Advanced Continuous Simulation Language (ACSL) User Guide/Reference Manual**, Mitchell & Gauthier Associates Inc., Version IV, Massachussets, USA, 1984.

3. Hermsen, P. A. **Ejection Characteristics of the BRU-33/A Vertical Ejector Rack (VER-2) Aboard a Static F/A-18 Aircraft**, Flight Mechanics Technical Memorandum 448, Aeronautical Research Laboratory, Melbourne, November 1991.

4. Fairlie, B. D. **Algorithms for the Reduction of Wind Tunnel Data from Strain Gauge Force Balances**, Aerodynamics Report 164, Aeronautical Research Laboratory, Melbourne, May 1984.

5. Fairlie, B. D., Roberts, L. J., Broadbent, R. G. **Transonic Wind Tunnel Tests on a 1/8th Scale Mk82 Bomb Model**, Aerodynamics Technical Memorandum 305, Aeronautical Research Laboratory, Melbourne, July 1977.

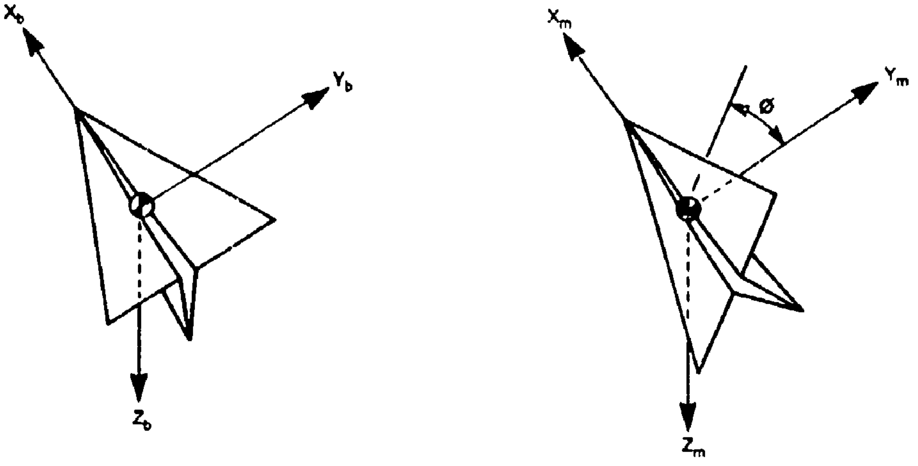


Figure 1: Body and missile axes systems.

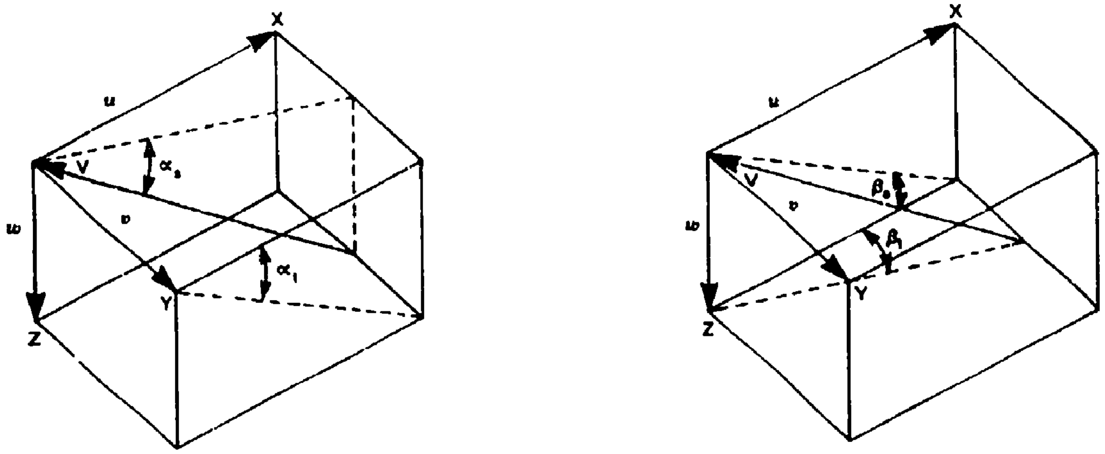


Figure 2: The difference between α_l, α_S and β_S, β_l.

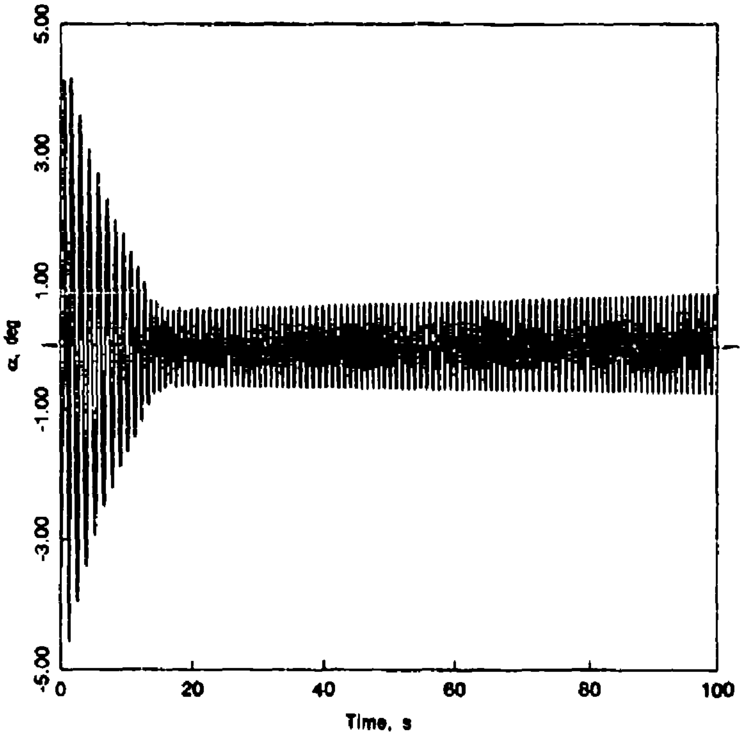


Figure 3: Result with gravity and linear forces set to zero, pure pitching moment.

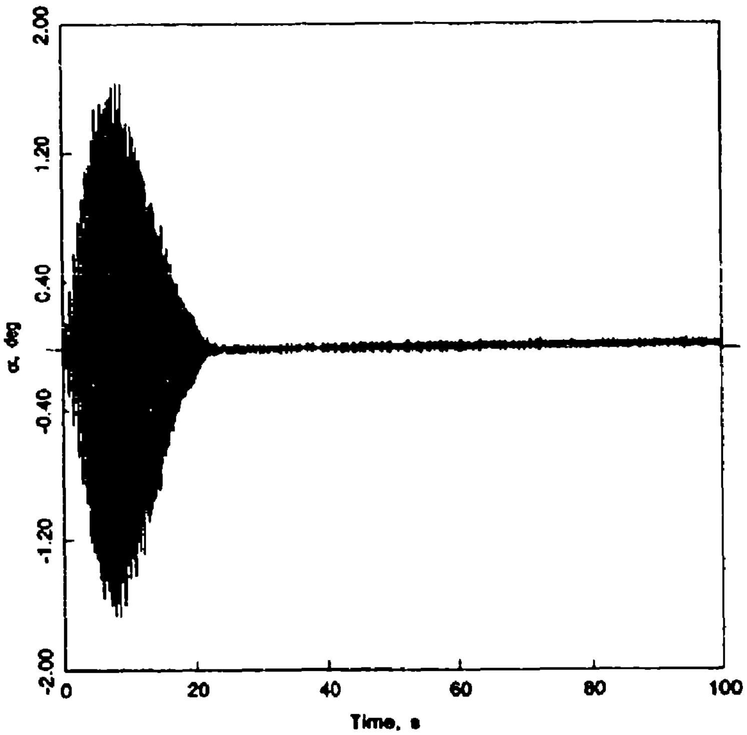


Figure 4: Result with gravity and linear forces set to zero, including all moments.

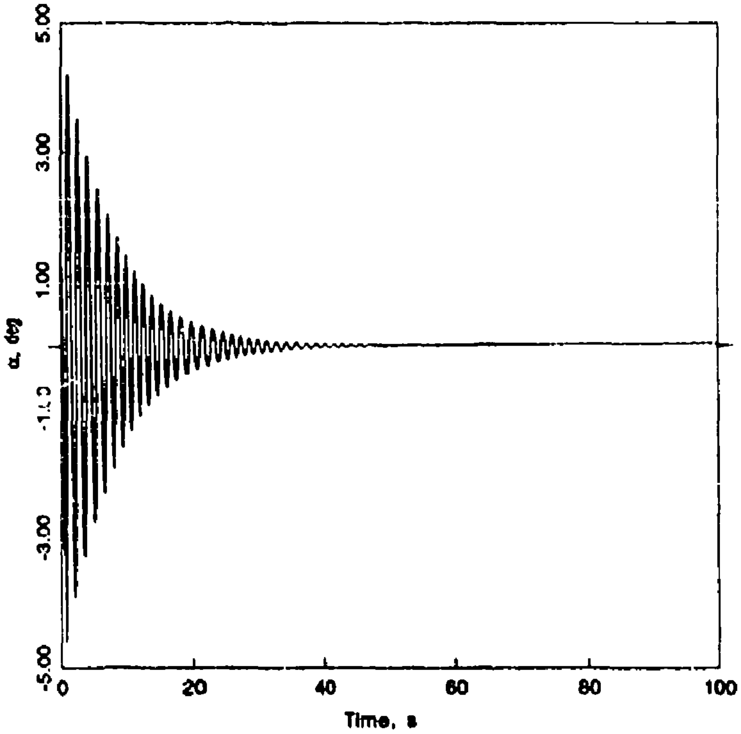


Figure 5: Result with gravity and linear forces set to zero, modified database, pure pitching moment.

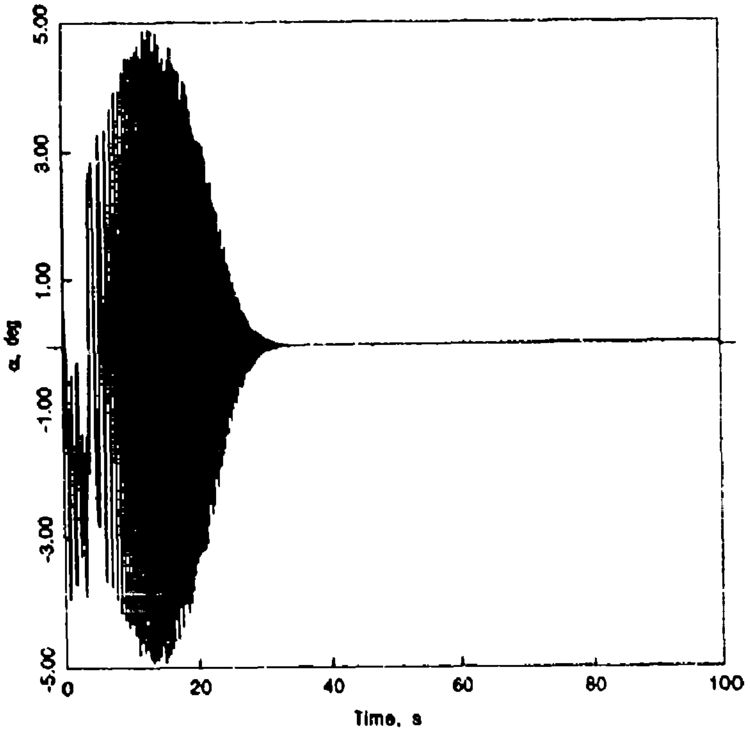


Figure 6: Result with gravity and linear forces set to zero, modified database, including all moments.

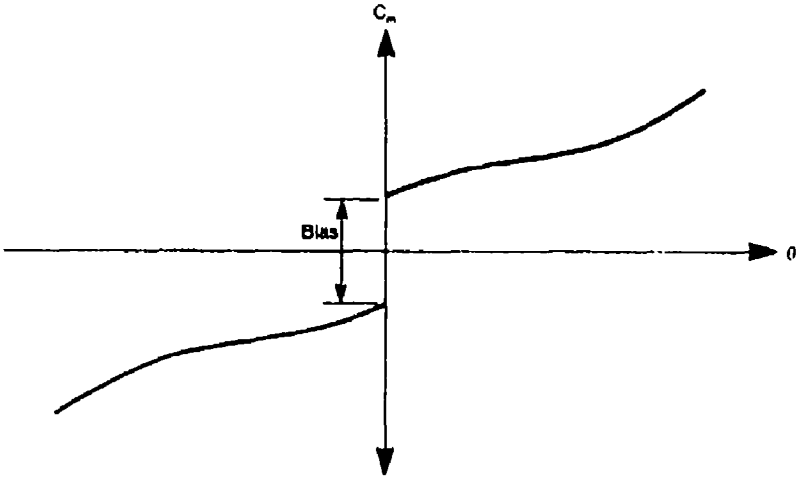


Figure 7: The data bias.

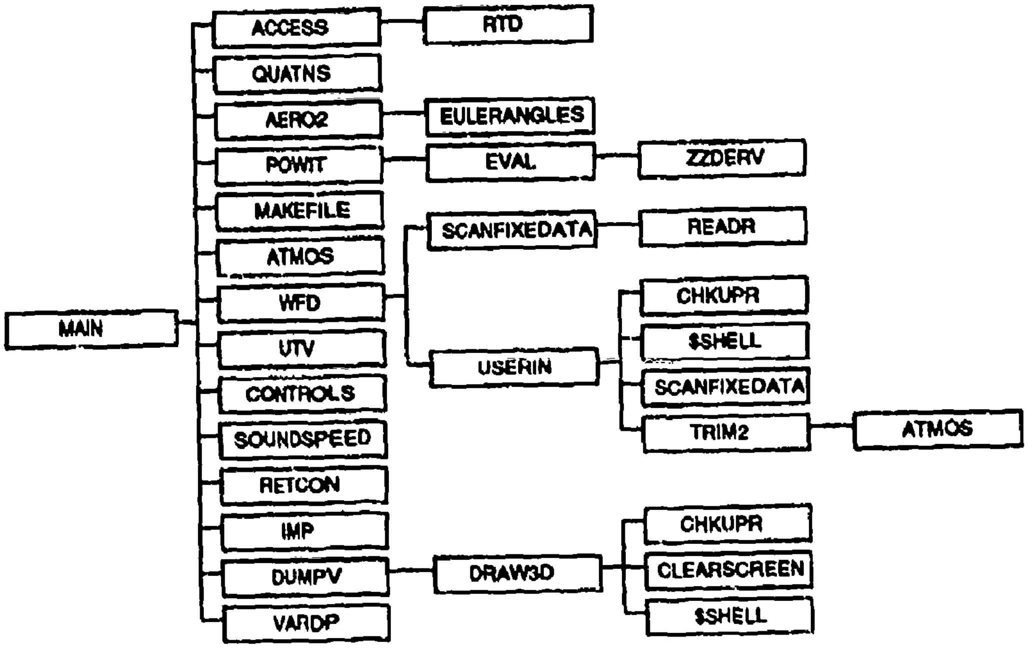


Figure 8: Program hierarchy.

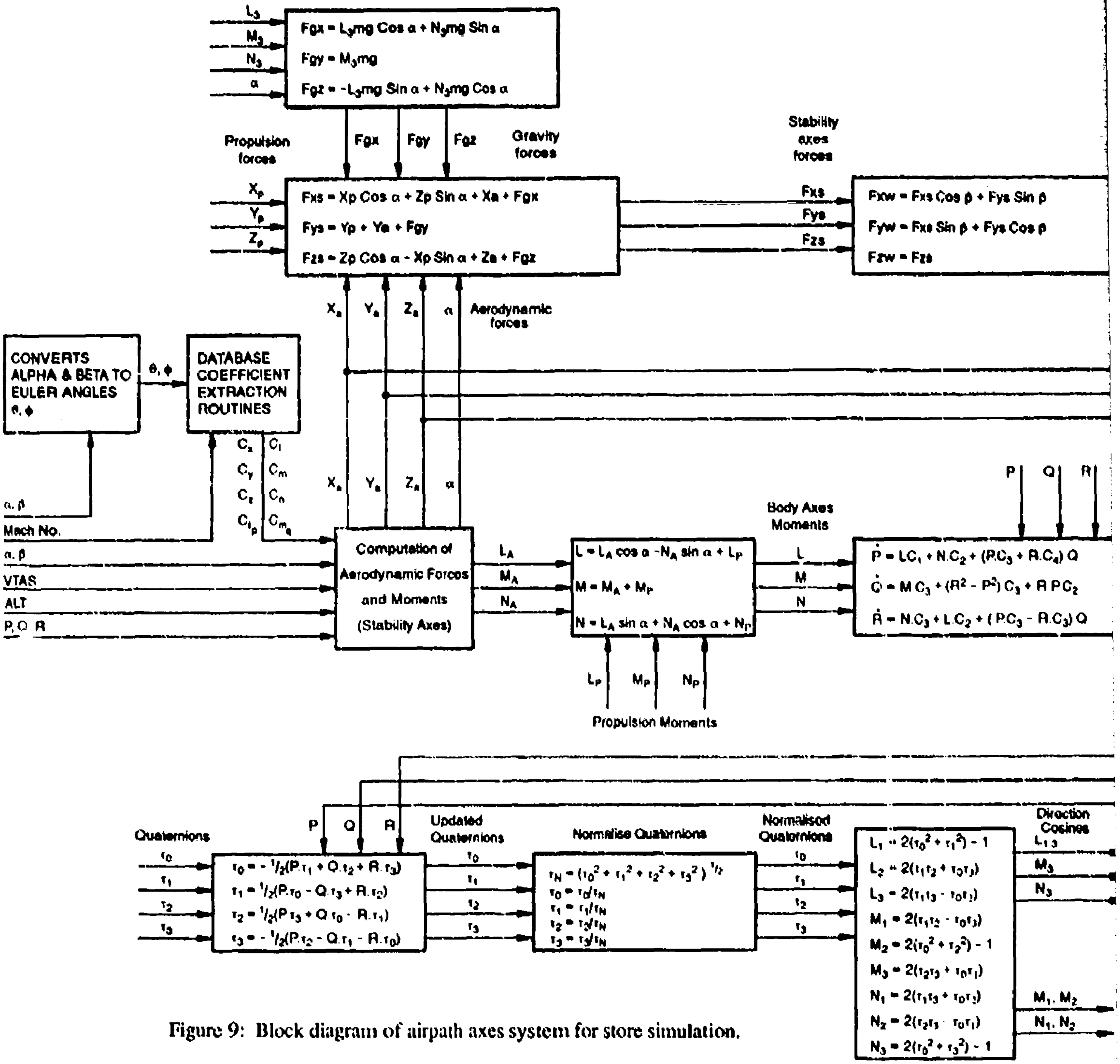
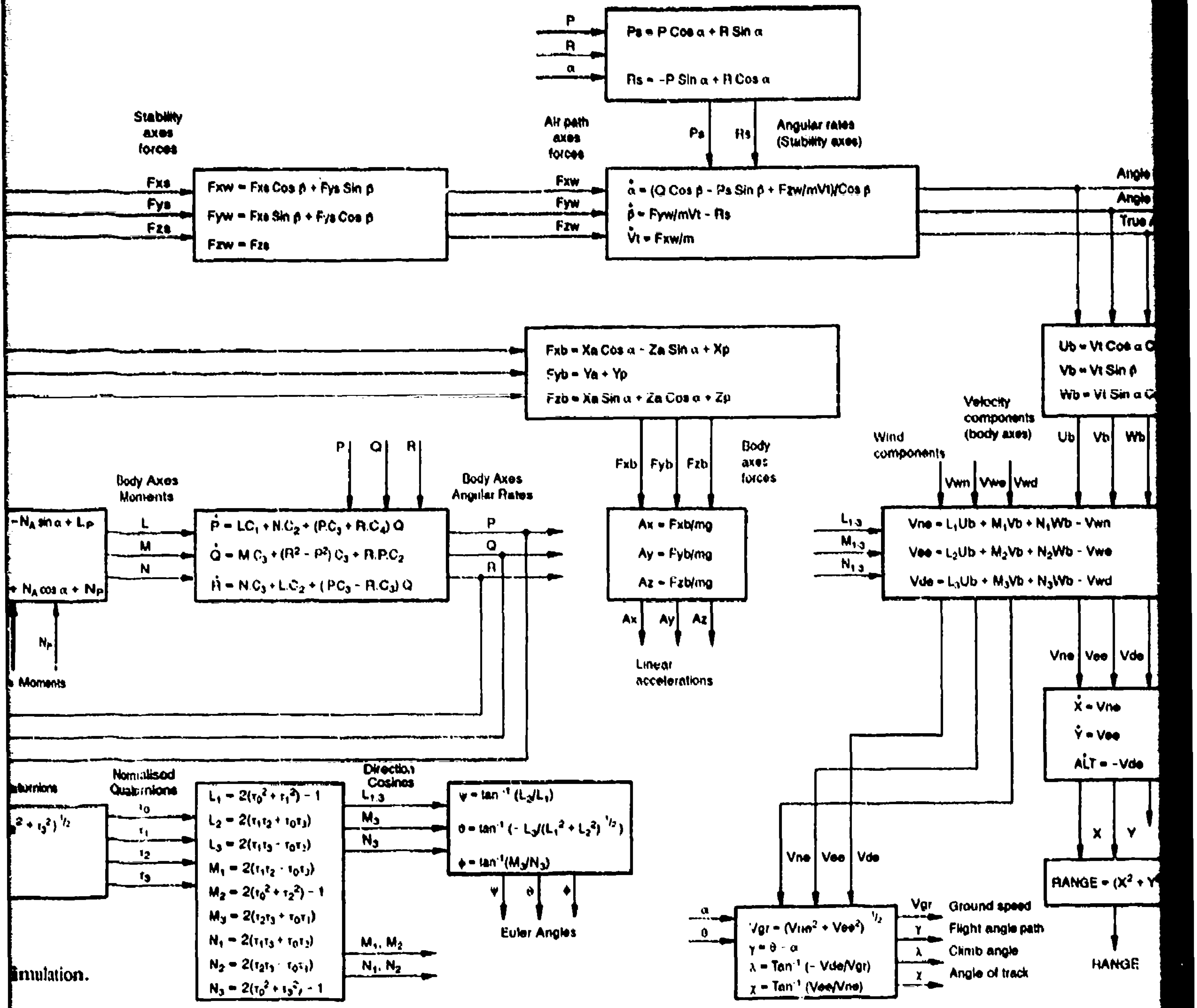
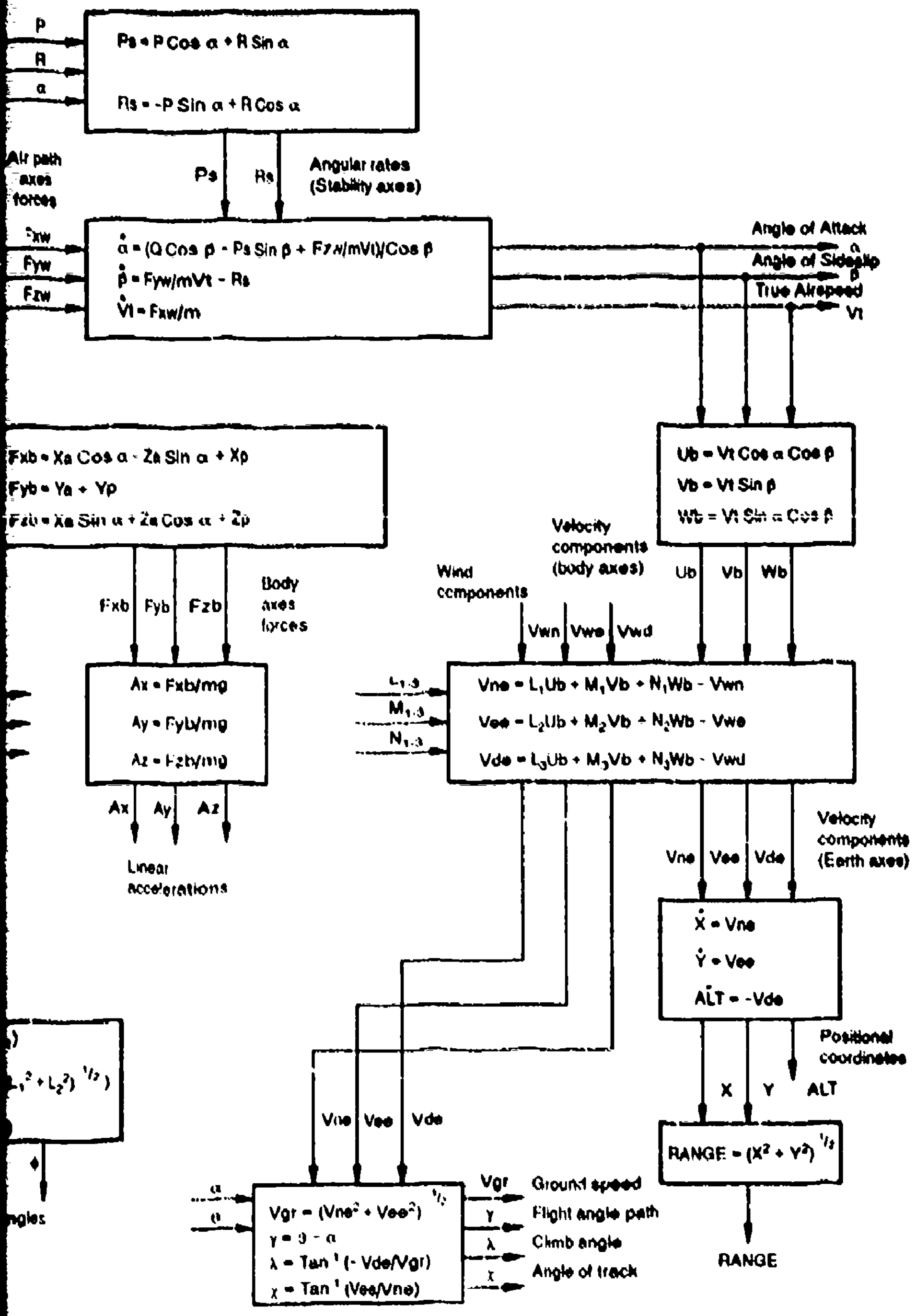

L_3
M_3
N_3
α
$Fgx = L_3 mg \cos\alpha + N_3 mg \sin\alpha$
$Fgy = M_3 mg$
$Fgz = -L_3 mg \sin\alpha + N_3 mg \cos\alpha$
Propulsion forces
Fgx
Fgy
Fgz
Gravity forces
Stability axes forces
X_p
Y_p
Z_p
$Fxs = Xp \cos\alpha + Zp \sin\alpha + Xa + Fgx$
$Fys = Yp + Ya + Fgy$
$Fzs = Zp \cos\alpha - Xp \sin\alpha + Za + Fgz$
Fxs
Fys
Fzs
$Fxw = Fxs \cos\beta + Fys \sin\beta$
$Fyw = Fxs \sin\beta + Fys \cos\beta$
$Fzw = Fzs$
X_a
Y_a
Z_a
α
Aerodynamic forces
CONVERTS ALPHA & BETA TO EULER ANGLES θ, ϕ
θ, ϕ
DATABASE COEFFICIENT EXTRACTION ROUTINES
C_x
C_y
C_z
C_{l_p}
C_l
C_m
C_n
C_{m_q}
X_a
Y_a
Z_a
α
α, β
Mach No.
α, β
VTAS
ALT
P, Q R
Computation of Aerodynamic Forces and Moments (Stability Axes)
L_A
M_A
N_A
$L = L_A \cos\alpha - N_A \sin\alpha + L_P$
$M = M_A + M_P$
$N = L_A \sin\alpha + N_A \cos\alpha + N_P$
L_P
M_P
N_P
Propulsion Moments
Body Axes Moments
L
M
N
P
Q
R
$\dot{P} = LC_1 + N.C_2 + (P.C_3 + R.C_4)\,Q$
$\dot{Q} = M.C_3 + (R^2 - P^2)\,C_3 + R\,P\,C_2$
$\dot{R} = N.C_3 + L.C_2 + (P.C_3 - R.C_3)\,Q$
Quaternions
τ_0
τ_1
τ_2
τ_3
P
Q
R
$\tau_0 = -\,{}^1/_2(P.\tau_1 + Q.\tau_2 + R.\tau_3)$
$\tau_1 = {}^1/_2(P.\tau_0 - Q.\tau_3 + R.\tau_2)$
$\tau_2 = {}^1/_2(P.\tau_3 + Q.\tau_0 - R.\tau_1)$
$\tau_3 = -\,{}^1/_2(P.\tau_2 - Q.\tau_1 - R.\tau_0)$
Updated Quaternions
τ_0
τ_1
τ_2
τ_3
Normalise Quaternions
$\tau_N = (\tau_0^2 + \tau_1^2 + \tau_2^2 + \tau_3^2)^{1/2}$
$\tau_0 = \tau_0/\tau_N$
$\tau_1 = \tau_1/\tau_N$
$\tau_2 = \tau_2/\tau_N$
$\tau_3 = \tau_3/\tau_N$
Normalised Quaternions
τ_0
τ_1
τ_2
τ_3
$L_1 = 2(\tau_0^2 + \tau_1^2) - 1$
$L_2 = 2(\tau_1\tau_2 + \tau_0\tau_3)$
$L_3 = 2(\tau_1\tau_3 - \tau_0\tau_2)$
$M_1 = 2(\tau_1\tau_2 - \tau_0\tau_3)$
$M_2 = 2(\tau_0^2 + \tau_2^2) - 1$
$M_3 = 2(\tau_2\tau_3 + \tau_0\tau_1)$
$N_1 = 2(\tau_1\tau_3 + \tau_0\tau_2)$
$N_2 = 2(\tau_2\tau_3 - \tau_0\tau_1)$
$N_3 = 2(\tau_0^2 + \tau_3^2) - 1$
Direction Cosines
$L_{1\,3}$
M_3
N_3
M_1, M_2
N_1, N_2

Figure 9: Block diagram of airpath axes system for store simulation.



P
R
α
$Ps = P \cos\alpha + R \sin\alpha$
$Rs = -P \sin\alpha + R \cos\alpha$
Ps
Rs
Angular rates (Stability axes)
Stability axes forces
Fxs
Fys
Fzs
$Fxw = Fxs \cos\beta + Fys \sin\beta$
$Fyw = Fxs \sin\beta + Fys \cos\beta$
$Fzw = Fzs$
Air path axes forces
Fxw
Fyw
Fzw
$\dot{\alpha} = (Q \cos\beta - Ps \sin\beta + Fzw/mVt)/\cos\beta$
$\dot{\beta} = Fyw/mVt - Rs$
$\dot{Vt} = Fxw/m$
Angle
Angle
True
$Fxb = Xa \cos\alpha - Za \sin\alpha + Xp$
$Fyb = Ya + Yp$
$Fzb = Xa \sin\alpha + Za \cos\alpha + Zp$
$Ub = Vt \cos\alpha C$
$Vb = Vt \sin\beta$
$Wb = Vt \sin\alpha C$
Velocity components (body axes)
Ub
Vb
Wb
Wind components
Vwn
Vwe
Vwd
Fxb
Fyb
Fzb
Body axes forces
P
Q
R
Body Axes Moments
Body Axes Angular Rates
$-N_A \sin\alpha + L_P$
$+ N_A \cos\alpha + N_P$
N_P
Moments
L
M
N
$\dot{P} = LC_1 + N.C_2 + (P.C_3 + R.C_4) Q$
$\dot{Q} = M C_3 + (R^2 - P^2) C_3 + R.P.C_2$
$\dot{R} = N.C_3 + L.C_2 + (P.C_3 - R.C_3) Q$
P
Q
R
$Ax = Fxb/mg$
$Ay = Fyb/mg$
$Az = Fzb/mg$
Ax
Ay
Az
Linear accelerations
$L_{1,3}$
$M_{1,3}$
$N_{1,3}$
$Vne = L_1Ub + M_1Vb + N_1Wb - Vwn$
$Vee = L_2Ub + M_2Vb + N_2Wb - Vwe$
$Vde = L_3Ub + M_3Vb + N_3Wb - Vwd$
Vne
Vee
Vde
$\dot{X} = Vne$
$\dot{Y} = Vee$
$\dot{ALT} = -Vde$
X
Y
$RANGE = (X^2 + Y$
RANGE
aternions
$^2 + \tau_3^2)^{1/2}$
Normalised Quaternions
τ_0
τ_1
τ_2
τ_3
$L_1 = 2(\tau_0^2 + \tau_1^2) - 1$
$L_2 = 2(\tau_1\tau_2 + \tau_0\tau_3)$
$L_3 = 2(\tau_1\tau_3 - \tau_0\tau_2)$
$M_1 = 2(\tau_1\tau_2 - \tau_0\tau_3)$
$M_2 = 2(\tau_0^2 + \tau_2^2) - 1$
$M_3 = 2(\tau_2\tau_3 + \tau_0\tau_1)$
$N_1 = 2(\tau_1\tau_3 + \tau_0\tau_2)$
$N_2 = 2(\tau_2\tau_3 - \tau_0\tau_1)$
$N_3 = 2(\tau_0^2 + \tau_3^2) - 1$
Direction Cosines
$L_{1,3}$
M_3
N_3
M_1, M_2
N_1, N_2
$\psi = \tan^{-1}(L_2/L_1)$
$\theta = \tan^{-1}(- L_3/(L_1^2 + L_2^2)^{1/2})$
$\phi = \tan^{-1}(M_3/N_3)$
ψ
θ
φ
Euler Angles
Vne
Vee
Vde
α
θ
$Vgr = (Vne^2 + Vee^2)^{1/2}$
$\gamma = \theta - \alpha$
$\lambda = \tan^{-1}(- Vde/Vgr)$
$\chi = \tan^{-1}(Vee/Vne)$
Vgr
γ
λ
χ
Ground speed
Flight angle path
Climb angle
Angle of track

imulation.


P
R
α
Ps = P Cos α + R Sin α
Rs = -P Sin α + R Cos α
Ps
Rs
Angular rates (Stability axes)
Air path axes forces
Fxw
Fyw
Fzw
α̇ = (Q Cos β - Ps Sin β + Fzw/mVt)/Cos β
β̇ = Fyw/mVt - Rs
V̇t = Fxw/m
Angle of Attack α
Angle of Sideslip β
True Airspeed Vt
Ub = Vt Cos α Cos β
Vb = Vt Sin β
Wb = Vt Sin α Cos β
Fxb = Xa Cos α - Za Sin α + Xp
Fyb = Ya + Yp
Fzb = Xa Sin α + Za Cos α + Zp
Fxb
Fyb
Fzb
Body axes forces
Wind components
Velocity components (body axes)
Vwn
Vwe
Vwd
Ub
Vb
Wb
Ax = Fxb/mg
Ay = Fyb/mg
Az = Fzb/mg
Ax
Ay
Az
Linear accelerations
L1-3
M1-3
N1-3
Vne = L1Ub + M1Vb + N1Wb - Vwn
Vee = L2Ub + M2Vb + N2Wb - Vwe
Vde = L3Ub + M3Vb + N3Wb - Vwd
Vne
Vee
Vde
Velocity components (Earth axes)
Ẋ = Vne
Ẏ = Vee
ALṪ = -Vde
Positional coordinates
X
Y
ALT
RANGE = (X² + Y²)^½
RANGE
L1² + L2²)^½)
ϕ
ngles
Vne
Vee
Vde
α
θ
Vgr = (Vne² + Vee²)^½
γ = θ - α
λ = Tan⁻¹ (- Vde/Vgr)
χ = Tan⁻¹ (Vee/Vne)
Vgr
γ
λ
χ
Ground speed
Flight angle path
Climb angle
Angle of track

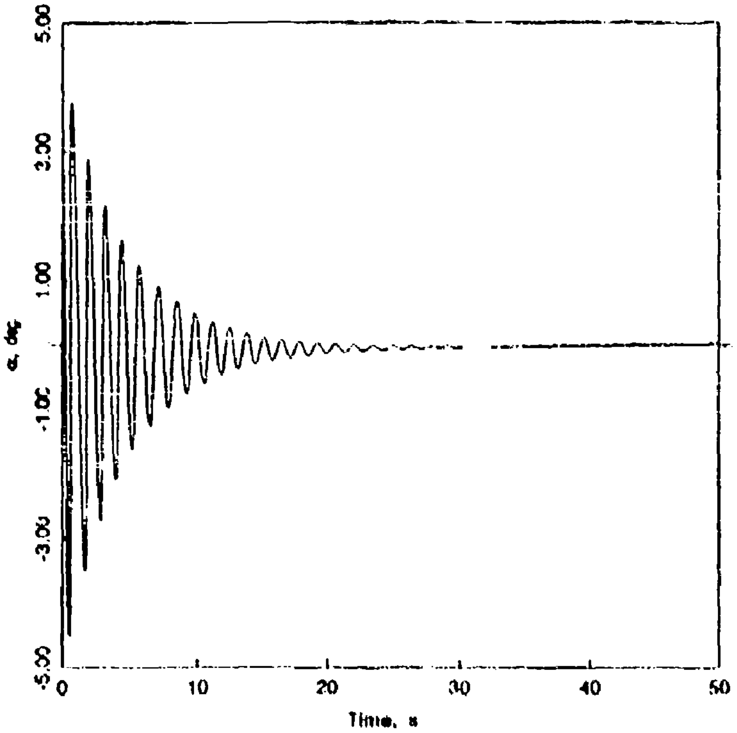


Figure 10: Time history of longitudinal model showing α.

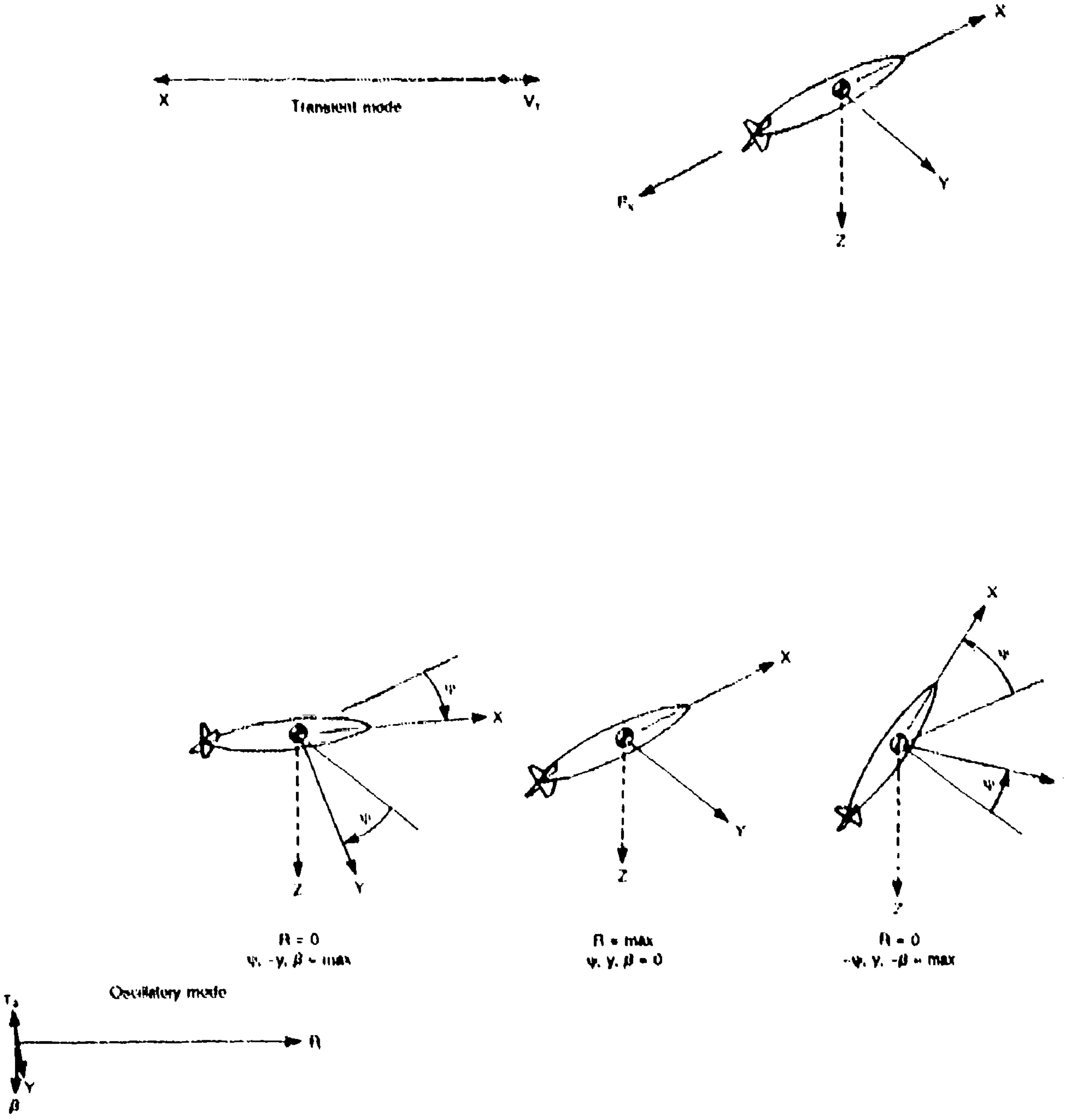


Figure 11: Eigen vectors and motion of longitudinal model.

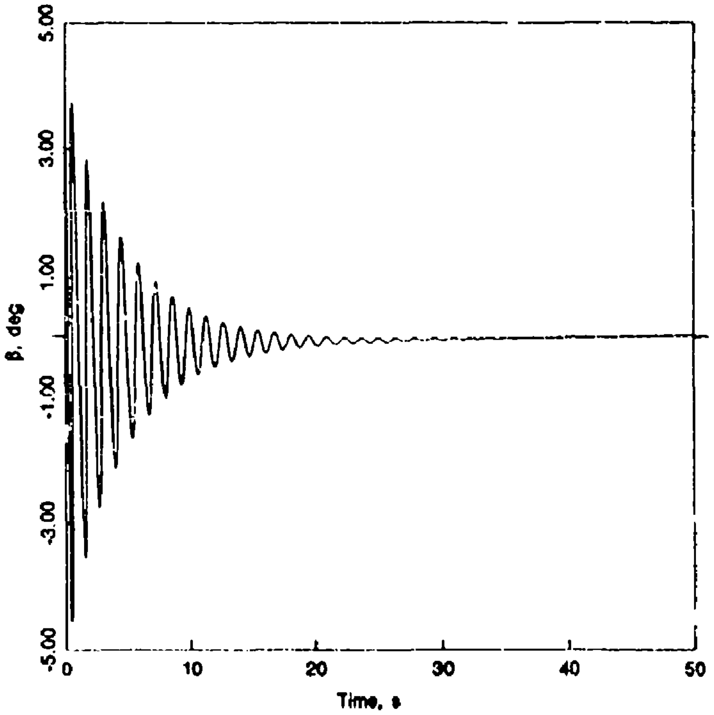


Figure 12: Time history of lateral model, showing β.

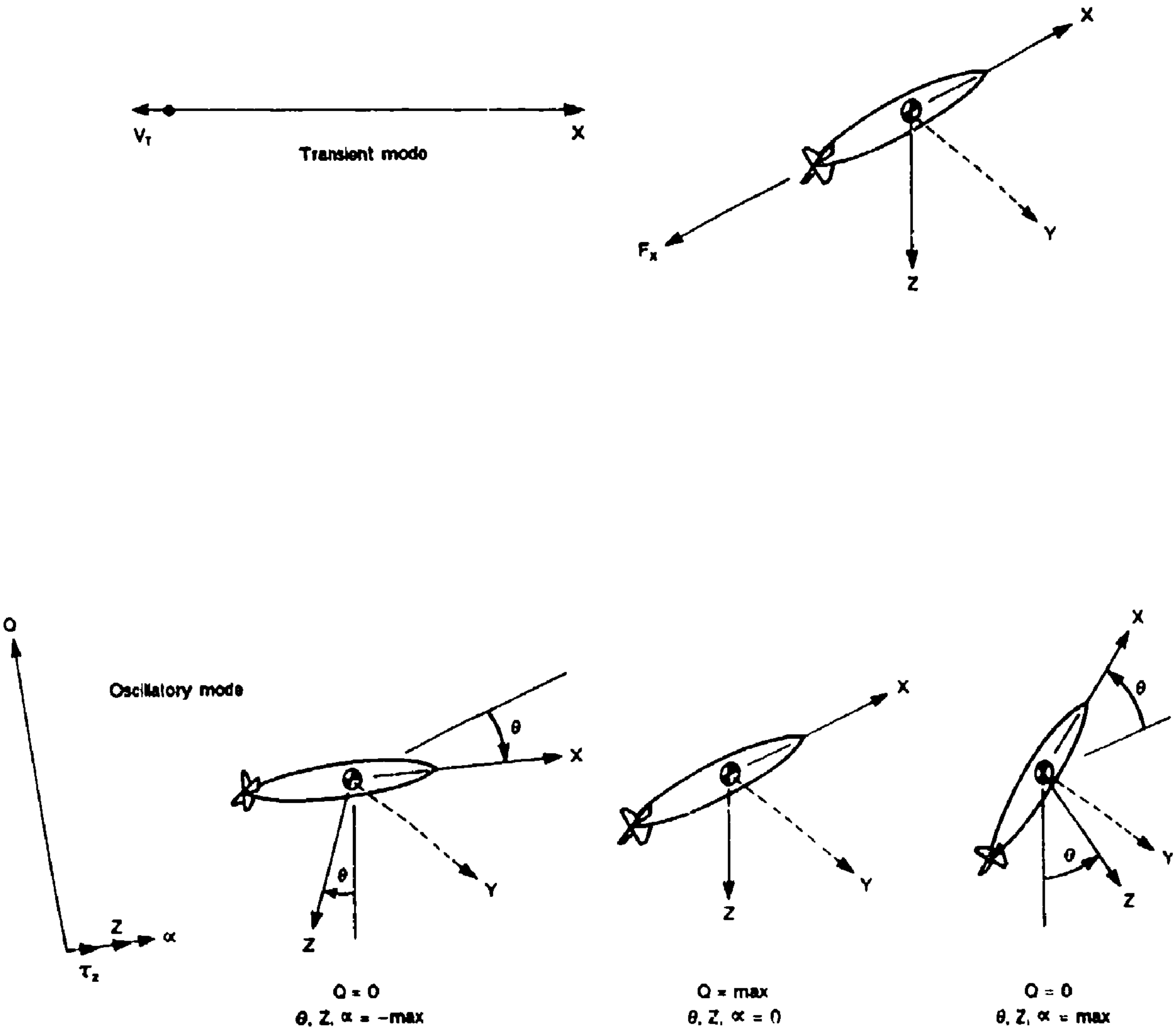


Figure 13: Eigen vectors and motion of lateral model.

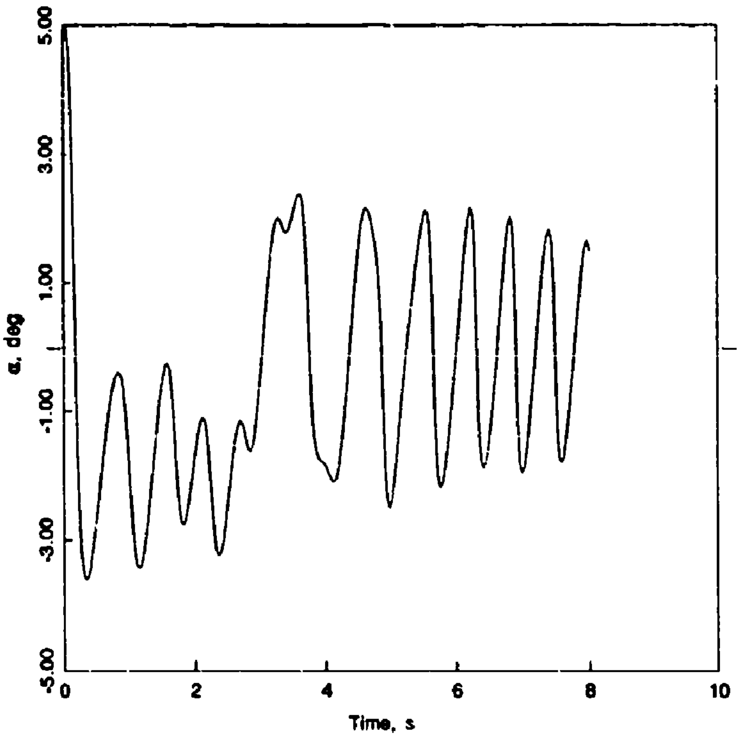
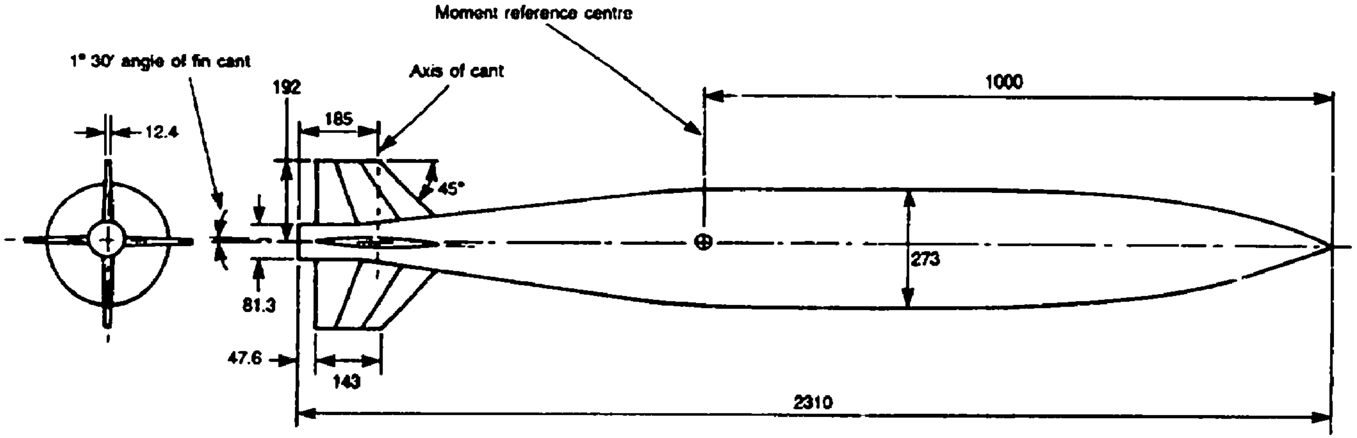


Figure 14: Typical time history of complete model, including gravity, showing α.



All dimensions in millimetres

Figure 15: The Mk-82 store geometry. (Reference 5)

Appendix A - The Geometric Properties of the Mk-82 Store

The geometric and mass data used in the simulation of the Mk-82 are as follows:

mass = 239.5 kg

I_{xx} = 2.35 $kg\ m^2$

I_{yy} = 56.74 $kg\ m^2$

I_{zz} = 56.74 $kg\ m^2$

CG from nose = 1.00 m

reference diameter = 0.273 m

The physical dimensions of the store are represented in Figure 15 (Reference 5).

Appendix B - Additional Subroutines Developed for Mk-82 Simulation

Subroutine AERO2

The AERO2 subroutine is called from the ACSL simulation program, with inputs of the angular rates p, q, r, and Mach number. The subroutine also has access via common blocks to the angle of attack and sideslip variables, AlphaR and BetaR. These variables are passed to a subordinate routine EULERANGLES which calculates the equivalent Euler attitude angles of pitch (Theta) and roll (Phi) used by the database. The appropriate force and moment coefficients are retrieved from the database using database routines, which have Machno, Phi, and Theta as inputs. The coefficients are then converted from missile axes to body axes via a rotation of Phi about the X - axis. Total body axes forces and moments are calculated as follows:

$$\text{XAR} = C_x.QBARS$$
$$\text{YAR} = C_y.QBARS$$
$$\text{ZAR} = C_z.QBARS$$
$$\text{LAR} = (C_l + \frac{C_{l_p}.b.p}{2.V}).QBARS.b$$
$$\text{MAR} = (C_m + \frac{C_{m_q}.b.q}{2.V}).QBARS.b$$
$$\text{NAR} = (C_n + \frac{C_{n_r}.b.r}{2.V}).QBARS.b\ ,$$

where C_{l_p}, C_{m_q}, and C_{n_r} are the damping derivatives in roll, pitch and yaw, QBARS is the dynamic pressure multiplied by the reference area ($\frac{\pi.b^2}{4}$), V is the true air speed of the store at a particular instant, and b is the reference diameter of the store, 0.273 m.

This subroutine has only one subordinate subroutine, called EULERANGLES.

Subroutines EJECT and EJECT2

The ACSL subroutine EJECT provides a simple representation of store ejection from the outboard station of the Vertical Ejector Rack. The Subroutine can be called on line from the ACSL prompt during the course of a simulation. The ejection of the store is simulated by setting the incremental forces and moments used in trimming the store to representative values for a specified duration to provide the correct impulse. The values used in EJECT are:

DZAR = 7828 N

DMAR = 0.0 N

DLAR = 0.0 N.m

Total Impulse = 940 N.s for t = 0.12 seconds.

EJECT2 emulates store release from the inboard station of the same Vertical Ejector Rack after station one has been fired. Reference 3 indicates that for store release from the F/A-18, ejection from the inboard station imparts initial roll and pitch rates to the store. The incremental forces and moments are reset to zero in the simulation after 0.12 seconds have elapsed. The values of the incremental forces and moments used are :

DZAR = 8200 N

DMAR = 19.8 N.m

DLAR = 94.3 N.m

Total Impulse = 984 N.s for t = 0.12 seconds.

These values correspond to the correct impulse for release given in Reference 3, and are intended to provide simple representations of store ejection characteristics.

Neither EJECT or EJECT2 have any subordinate subroutines.

Subroutine MAKEFILE

The MAKEFILE subroutine is used to record the trajectory and orientation of the store in ASCII format for each simulation run. The subroutine uses standard FORTRAN code to open (or reopen) a file at the beginning of a run, overwriting any existing contents of the file with the current trajectory data. The information recorded is the current time, x,y,z positions, yaw, pitch, roll angles, angles of attack and sideslip, airspeed, rate of change of altitude, and normal acceleration, with the following format:

```
''write(1,1000(E,' '))time, x, y, -z, Psid, Thetad, Phid, Alphad, Betad,
Vt, vde, AN, 0, 0, 0 '',
```

where the last three zeroes are unused channels.

This file can be used to obtain the position, speed, and attitude of the store in any post-run processing that is required.

The first time that the MAKEFILE subroutine is called, the user is prompted with the screen message:

```
"At what time would you like to begin the datafile? ",
```

at which point the program awaits a numerical input from the user. This is for the purpose of [illegible] the store trajectory data to part of the flight path of an aircraft, so that the store and aircraft coincide until the time specified by the user for the store simulation to begin. The file is overwritten each time a run is commenced, so that only the final simulation run is recorded. The output file is called STORE.DAT.

Note: For viewing the recorded trajectory, a graphics replay system has been developed using an IRIS graphics work station. The data file STORE.DAT can be transferred to the IRIS workstation and the trajectory of the store can be viewed using the graphics replay program.

The MAKEFILE subroutine has no subordinate subroutines.

Subroutine SOUNDSPEED

This subroutine is called from the main ACSL program, with the current altitude as input, and speed of sound at that altitude as output. This value is then used to calculate the Mach number which is required by the AERO2 subroutine and the specialised database extraction subroutines.

The SOUNDSPEED subroutine has no subordinate subroutines.

Subroutine EULERANGLES

This subroutine is called from AERO2 and has angle of attack and sideslip as its inputs. The outputs are the Euler angles Theta and Phi, with Psi being set at zero. The Euler angles describe the orientation in space of a body relative to an inertial reference frame, the Earth-Axes. The relevant formulae are reproduced here (see Reference 4) :

$$cos\theta = cos(\alpha)cos(\beta) \text{ and}$$

$$tan\phi = \frac{tan(\beta)}{sin(\alpha)} \ .$$

The correct sign of θ is determined by testing for the sign of α, and equating the sign of θ.

The angles of roll (Phi) and pitch (Theta) are returned to the AERO2 subroutine, where they are needed as input to the database access routines, along with the Mach number.

The subroutine EULERANGLES has no subordinate subroutines.

Subroutine ACCESS

The subroutine ACCESS is called from the beginning of the main ACSL simulation program. The purpose of ACCESS is to provide a platform for the calling of the database routine, RTD, which opens and prepares the database for retrieval of specific data.

The subroutine ACCESS has the subroutine RTD as a subordinate subroutine, which is part of the database extraction routines.

Appendix C - Database Conversion Program

The database conversion program was developed to transform the format of the original database into the format required by the standard database routines. The conversion was achieved by reading the values of the variables from the original database which is stored as an ASCII file on a 3.5 inch floppy disk, and rewriting them to a new file in the required format. Included in this Appendix is a copy of the Database Conversion program.

```
      program convertformat
      real  half1(12),half2(5),line(15),lump(16,13,13,8)
      integer i,j,k,l,p,q,r,count

c     lump(a,a,a,1,2,3,4,5,6,7,8) refer to cx,cy,cz,cl,cm,cn,clp,cmq
      print*,'Reading Data!'
      open(10,file='mk82cx.dat',status='old',action='read')
        do 20 l=1,2704,1
        read(10,100)(half1(i),i=1,12)
        read(10,110)(half2(j),j=1,5)

          do 10 k=1,15,1
          if(k.le.12)line(k)=half1(k)
10        if(k.gt.12)line(k)=half2(k-12)

          lump(line(2),line(3),line(4),1)=line(8)
          lump(line(2),line(3),line(4),2)=line(9)
          lump(line(2),line(3),line(4),3)=line(10)
          lump(line(2),line(3),line(4),4)=line(11)
          lump(line(2),line(3),line(4),5)=line(12)
          lump(line(2),line(3),line(4),6)=line(13)
          lump(line(2),line(3),line(4),7)=line(14)
          lump(line(2),line(3),line(4),8)=line(15)

20    continue

      open(11,file='mk82db.dat',status='new',action='write',
     &carriagecontrol='fortran')


      print*,'CX     POINTS'


      write(11,150)
      write(11,160)
      write(11,170)
      write(11,180)
      write(11,190)
      write(11,200)
      write(11,210)
      write(11,220)
      write(11,230)
      write(11,240)
      write(11,250)
      write(11,160)
      write(11,270)
      write(11,280)
      write(11,290)

      write(11,160)
      write(11,330)
      write(11,115)
      print*,'Writing to file'

      print*,'Doing CX.'
      count=0
        do 30 r=1,13,1
          do 31 q=1,13,1
            do 32 p=1,16,1
              write(11,120)lump(p,q,r,1)
              count=count+1
              if(count.eq.5)then
                write(11,115)
                count=0
              endif
 32     continue
 31     continue
 30     continue
        write(11,320)
        print*,'Finished CX.'

        print*,'Doing CY.'
        write(11,340)
        write(11,115)
        count=0
          do 33 r=1,13,1
            do 34 q=1,13,1
              do 35 p=1,16,1
                write(11,120)lump(p,q,r,2)
                count=count+1
                if(count.eq.5)then
                  write(11,115)
                  count=0
                endif
 35     continue
 34     continue
 33     continue
        write(11,320)
        print*,'Finished CY.'

        print*,'Doing CZ.'
        write(11,350)
        write(11,115)
        count=0
          do 36 r=1,13,1
            do 37 q=1,13,1
              do 38 p=1,16,1
                write(11,120)lump(p,q,r,3)
                count=count+1
                if(count.eq.5)then
                  write(11,115)
                  count=0
                endif
 38     continue
 37     continue
 36     continue
        write(11,320)
        print*,'Finished CZ.'

        print*,'Doing C1.'
        write(11,360)
        write(11,115)
        count=0
          do 39 r=1,13,1
            do 40 q=1,13,1
              do 41 p=1,16,1
                write(11,120)lump(p,q,r,4)
                count=count+1
                if(count.eq.5)then
                 write(11,115)
                  count=0
                endif
 41     continue
 40     continue
 39     continue
        write(11,320)
```

```
      print*,'Finished Cl.'

      print*,'Doing CM.'
      write(11,370)
      write(11,115)
      count=0
        do 42 r=1,13,1
          do 43 q=1,13,1
            do 44 p=1,16,1
               write(11,120)lump(p,q,r,5)
               count=count+1
               if(count.eq.5)then
                 write(11,115)
                 count=0
               endif
44    continue
43    continue
42    continue
      write(11,320)
      print*,'Finished CM.'

      print*,'Doing CN.'
      write(11,380)
      write(11,115)
      count=0
        do 45 r=1,13,1
          do 46 q=1,13,1
            do 47 p=1,16,1
               write(11,120)lump(p,q,r,6)
               count=count+1
               if(count.eq.5)then
                 write(11,115)
                 count=0
               endif
47    continue
46    continue
45    continue
      write(11,320)
      print*,'Finished CN.'

      print*,'Doing Clp.'
      write(11,390)
      write(11,115)
      count=0
        do 48 r=1,13,1
          do 49 q=1,13,1
            do 50 p=1,16,1
               write(11,120)lump(p,q,r,7)
               count=count+1
               if(count.eq.5)then
                write(11,115)
                 count=0
               endif
50    continue
49    continue
48    continue
      write(11,320)
      print*,'Finished Clp.'

      print*,'Doing Cmq.'
      write(11,400)
      print*,'OK1'
      write(11,115)
      print*,'OK2'
      count=0
      print*,'OK3'
        do 51 r=1,13,1
          do 52 q=1,13,1
            do 53 p=1,16,1
            write(11,120)lump(p,q,r,8)
              count=count+1
               if(count.eq.5)then
                 write(11,115)
                 count=0
               endif
53    continue
52    continue
51    continue
      write(11,320)
      print*,'Finished Cmq.'

100   format(4i5,1f15.4,7f13.4)
110   format(31x,5f13.4)
115   format('0')
120   format('&',1f13.4,',')

150   format('           title  mk82cxdatabase')
160   format(' *')
170   format('          real  cx,cy,cz,cl,cm,cn,clp,cmq')
180   format('          thruput cx    (alpha,roll,machno)')
190   format('          thruput cy    (alpha,roll,machno)')
200   format('          thruput cz    (alpha,roll,machno)')
210   format('          thruput cl    (alpha,roll,machno)')
220   format('          thruput cm    (alpha,roll,machno)')
230   format('          thruput cn    (alpha,roll,machno)')
240   format('          thruput clp   (alpha,roll,machno)')
250   format('          thruput cmq   (alpha,roll,machno)')
260   format(' *')

270   format(' alpha  conbpt 0.0,30.0,2/')
280   format(' roll   conbpt -45.0,45.0,7.5/')
290   format(' machno varbpt 0.4,0.5,0.6,0.7,0.8,0.85,0.9,0.95,1.0,1.05
     &,1.1,1.15,1.2/')
300   format(' *')

320   format('                  ++')
330   format(' CX     POINTS')
340   format(' Cy     POINTS')
350   format(' Cz     POINTS')
360   format(' Cl     POINTS')
370   format(' Cm     POINTS')
380   format(' Cn     POINTS')
390   format(' Clp    POINTS')
400   format(' Cmq    POINTS')

      stop
      end
```

Appendix D - Example Simulation Run

Run with Clean Release

In this first example, the store is released intc the free stream without the benefit of any ejection forces or moments. The operator input is shown here in lower case, but it should be noted that the program is not case sensitive. The computer echo of the inputs are shown in upper case. The source code file is named adm.acsl, with the setup file named adm.setup. The executable file is called adm.exe. To run the program, type

1. adm

   The program prompts with

   ```
   Roading data.....ploaso hold
   ```

   The program then scans the breakpoint data, and reads the points CX, CY, . . . CMQ. When CMQ has been read, the cursor awaits the next user input.

2. s cmd=10

   s cmd=10 instructs the program to take its input from the setup file, until instructed otherwise by a statement contained within the input file.

   The pure ACSL segment of the program paraphrases each instruction on the screen, coming back with S CMD=10.

3. s x0=1000

   If the simulation output file STORE.DAT is to be used in a flight path reconstruction in conjunction with another aircraft, the store trajectory must begin at the down range distance of the aircraft at the time of release, so that the store does not leap back to X=0.

4. odit

   The EDIT subroutine is not a standard ACSL command, but allows the setting of the aircraft parameters and release conditions. The aircraft parameters are all preset to values reflecting the characteristics of the Mk-82 store, and the release conditions are set to default values which need to be modified for each different release.

5. alt0=1000

   ALT0 is the initial altitude in metres of the store. Note that the values within EDIT are input differently to those values input in the main ACSL program.

6. q0=0.01

   q0 is the initial pitch rate experienced by the store, in radians per second.

7. vok0=300

vek0 is the initial velocity of the store at the point of release, in knots.

8. `thetd0=10`

thetd0 is the initial pitch attitude of the store, in degrees.

9. `alphd0=2`

alphd0 is the initial angle of attack of the store, in degrees.

10. `betad0=2`

betad0 is the initial angle of sideslip of the store, in degrees.

11. `trim`

The first time this routine is called, the user is prompted with the question, "`At what time would you like to begin the data file?`" The program awaits the input of a number that represents the time in seconds that the parent aircraft flies before the store is released. This is only relevant if the flight graphic replay is going to be used in conjunction with another aircraft. If the store is intended to be visualised exclusively, the start time can be set to zero. Note that this value affects only the STORE.DAT file which is used in later graphic replay, and not the simulation.

The only reason that this store is trimmed is so that the EIGEN function is applied to a body in equilibrium. TRIM introduces artificial incremental forces and moments which negate all accelerations experienced by the store at the point of release.

EDIT has now been exited, and the user is returned to the pure ACSL environment.

12. `eigen`

EIGEN performs an eigen analysis, producing eigen values, eigen vectors, and the Jacobian matrix. These may be printed out by inputting the statement '`s prn=9` ' prior to the input of the EIGEN statement. This causes the output of the EIGEN routine to be diverted to an output file, ADM.L , which may then be printed out with the command PRINT ADM.L after the program has been exited.

13. `dtz`

To remove the incremental forces and moments quickly, they can all be set to zero with the statement DTZ, which simply stands for "Deltas to zero".

14. `s tstop=10`

The length of the simulation run is set by the statement,

```
s tstop=10
```

, where the time is entered in seconds.

15. go

GO begins the simulation.

16. d alt

The statement 'd alt' displays the final value of the specified variable. In this case, the final altitude is 98 feet, indicating that the run time is too short. Repeat the last three statements iteratively until satisfied with the final altitude of the store.
When complete, type

```
stop
```

or Control-c, which will exit the ACSL program. The STORE.DAT file contains only the final simulation run, and will be overwritten every time ADM is run, so if the file is needed, it is necessary to copy STORE.DAT to a more appropriate file name.
If

```
s prn=9
```

has been set, ADM.L will contain the results of the eigen analysis, which may then be printed.

Run With Ejection Release

To run a simulation including simulated ejection forces and moments, the user must invoke one of the ejection routines after the store has been trimmed and all the incremental forces set to zero (step 13). The procedure EJECT simulates the ejection forces and moments acting on a store released from one station on a vertical ejector rack, and EJECT2 simulates ejection forces and moments from a second station on the same rack, after the first store is released. The applied impulse is different in both cases due to the different stiffnesses and inertias encountered by the rack in each case, resulting in different forces and moments being applied (see Appendix B.2 for details of ejection impulses). The actual ejection of the store is completed in 0.12 seconds, so the ACSL program tests for time greater than 0.12 seconds, and sets the incremental forces to zero when this condition is true. Because of this, it is necessary to call EJECT or EJECT2 each time the simulation is run, ie before the command GO is input.

Note: if it is necessary to trim the store again during the course of a simulation study, the program will crash and exit unless at least one of the incremental forces or moments is reset to a non zero value to prevent a division by zero.

DISTRIBUTION

AUSTRALIA

DEFENCE ORGANISATION

Defence Science and Technology Organisation

Chief Defence Scientist }
FAS Science Policy } shared copy
AS Science Corporate Management }
Counsellor Defence Science, London (Doc Data Sheet only)
Counsellor Defence Science, Washington (Doc Data Sheet only)
Senior Defence Scientific Adviser (Doc Data Sheet only)
Scientific Adviser Policy and Command (Doc Data Sheet only)
Navy Scientific Adviser (3 copies Doc Data Sheet)
Scientific Adviser - Army (Doc Data Sheet only)
Air Force Scientific Adviser
Director Trials

Aeronautical Research Laboratory
Director
Library
Chief Air Operations Division
Author: M. Shilo
C.A. Martin
J.S. Drobik
B.D. Fairlie

Materials Research Laboratory
Director/Library

Main Library - DSTO Salisbury

Electronics Research Laboratory
GWD
C. Jermey

Defence Central

OIC TRS, Defence Central Library
Document Exchange Centre, DIS (8 copies)
Defence Intelligence Organisation
Library, Defence Signals Directorate (Doc Data Sheet Only)

Air Force
Aircraft Research and Development Unit
Scientific Flight Group
Library
OIC ATF, ATS, RAAFSTT, Wagga (2 copies)

HQ ADF

Director General Force Development (Air)

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy

Library

Head of Aerospace and Mechanical Engineering

Other Organisations

NASA (Canberra)

AGPS

SPARES (3 COPIES)

TOTAL (35 COPIES)

AL 149 DEPARTMENT OF DEFENCE

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

DOCUMENT CONTROL DATA

1a. AR NUMBER AR-006-666	1b. ESTABLISHMENT NUMBER ARL-TN-25	2. DOCUMENT DATE JANUARY 1994	3. TASK NUMBER ADA 91/043

4. TITLE

SIX DEGREE OF FREEDOM FLIGHT DYNAMIC MODEL OF A MK-82 STORE

5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF. (C) RESTRICTED (R), LIMITED (L), UNCLASSIFIED (U))

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES
38

7. NO. REFS.
5

8. AUTHOR(S)

M.A. SHILO

9. DOWNGRADING/DELIMITING INSTRUCTIONS

Not applicable.

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY
AIR OPERATIONS DIVISION
506 LORIMER STREET
FISHERMENS BEND VIC 3207

11. OFFICE/POSITION RESPONSIBLE FOR:

SPONSOR RAAF DGFD (AIR)/DCFD
SECURITY -
DOWNGRADING -
APPROVAL CAOD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)

Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO ....

No limitations.

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE ☒ UNRESTRICTED OR ☐ AS FOR 13a.

14. DESCRIPTORS
Mk-82 bombs
Free flight
Flight dynamics
External store separation

15. DISCAT SUBJECT CATEGORIES
1902

16. ABSTRACT

A six degree of freedom flight dynamic model incorporating an aerodynamic database derived from DSTO wind tunnel data was developed for the Mk-82 store. The model can be linked with aircraft flight dynamic models, and forms part of a general programme to model aircraft carriage and delivery, weapon release and store ballistics. A rudimentary ejection model was also incorporated to represent the forces and moments experienced by the store at the point of release. The model is valid for free stream flight only, and ignores the effects of aircraft-store interaction.

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT).

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER Technical Note 25	19. WA NUMBER 52 511A	20. TYPE OF REPORT AND PERIOD COVERED

21. COMPUTER PROGRAMS USED

Advanced continuous simulation language (ACSL)

22. ESTABLISHMENT FILE REF(S)

S262

23. ADDITIONAL INFORMATION (AS REQUIRED)